Direct tensor-product solution of one-dimensional elliptic equations with parameter-dependent coefficients

Sergey V. Dolgov[a,1], Vladimir A. Kazeev[b,1], Boris N. Khoromskij[c]

[a] *University of Bath, Claverton Down, BA2 7AY Bath, United Kingdom.*
[b] *Department of Mathematics, Stanford University. 450 Serra Mall, 94305 Stanford, USA.*
[c] *Max-Planck Institute for Mathematics in the Sciences. Inselstr. 22-26, 04103 Leipzig, Germany.*

Abstract

We consider a one-dimensional second-order elliptic equation with a high-dimensional parameter in a hypercube as a parametric domain. Such a problem arises, for example, from the Karhunen–Loève expansion of a stochastic PDE posed in a one-dimensional physical domain. For the discretization in the parametric domain we use the collocation on a tensor-product grid. The paper is focused on the tensor-structured solution of the resulting multiparametric problem, which allows to avoid the curse of dimensionality owing to the use of the separation of parametric variables in the tensor train and quantized tensor train formats.

We suggest an efficient tensor-structured preconditioning of the entire multiparametric family of one-dimensional elliptic problems and arrive at a direct solution formula. We compare this method to a tensor-structured preconditioned GMRES solver in a series of numerical experiments.

Keywords: elliptic equations, parametric problems, iterative methods, tensor formats, Sherman-Morrison correction, preconditioning
2010 MSC: 35J15, 15A69, 65F10, 34B08, 60H15

Email addresses: S.Dolgov@bath.ac.uk (Sergey V. Dolgov), kazeev@stanford.edu (Vladimir A. Kazeev), bokh@mis.mpg.de (Boris N. Khoromskij)

[1]The research presented in this paper was carried out while S. Dolgov was working at INM RAS (Moscow) and MPI MiS (Leipzig) and while V. Kazeev was working at INM RAS and visiting MPI MiS. At the Univerity of Bath, S. Dolgov was supported by the EPSRC fellowship EP/M019004/1.

1. Introduction

We consider a multiparametric one-dimensional linear elliptic problem posed in the physical domain $D = (0, 1)$ and the parameter domain $\boldsymbol{Y} = [-1, 1]^M$:

$$\begin{aligned} -\frac{\partial}{\partial x} a(x, y) \frac{\partial u(x, y)}{\partial x} &= f(x, y), \qquad x \in D, \quad y \in \boldsymbol{Y}, \\ u &= 0, \qquad x \in \partial D, \quad y \in \boldsymbol{Y}. \end{aligned} \tag{1.1}$$

Here, y denotes a tuple of parameters, $y = (y_1, \ldots, y_M) \in \boldsymbol{Y}$. For the solution of (1.1), we use the classical Finite Element Method (FEM) for the discretization in the physical domain D, and the collocation method [1] in the parameter domain $\boldsymbol{Y}$. More details are given in Section 3.1. For every $y \in \boldsymbol{Y}$ the problem (1.1) reduces to the form

$$\begin{aligned} -\frac{d}{dx} a(x) \frac{du(x)}{dx} &= f(x), \qquad x \in D, \\ u &= 0, \qquad x \in \partial D. \end{aligned} \tag{1.2}$$

If we consider a standard FEM discretization of (1.2) with N degrees of freedom, it involves a tridiagonal $N \times N$-matrix, and the discrete problem is solved easily by the Gaussian elimination algorithm through $\mathcal{O}(N)$ operations.

Assume that we apply the same discretization to (1.1) and also discretize the problem with respect to the parameters by choosing a finite set of collocation points $y_s \in \boldsymbol{Y}$, $s = 1, \ldots, S$ and considering a set of problems (1.2) with $y = y_s$, $s = 1, \ldots, S$. This results in a large linear system with a block-diagonal matrix, which consists of S blocks, each of them being the tridiagonal matrix corresponding to a fixed parameter $y = y_s$. The direct methods cannot be applied to such a discretization straightforwardly for the solution of the whole family of problems, parameterized by a high-dimensional parameter $y \in \boldsymbol{Y}$. The reason for that is that the elementwise representation of the data itself suffers from the so-called "curse of dimensionality" with respect to M, which can be beyond tens or hundreds. For example, if the problem (1.1) approximates a stochastic PDE, the dimensionality M of the parametric space is governed by the accuracy of this approximation and, thus, may need to be high.

In the present paper we use the collocation on tensor-product uniform grids in $\boldsymbol{Y}$ with $S = n^M$ and employ the tensor-structured representation of the discrete problem to extract the "effective" degrees of freedom adaptively and make the problem tractable. A broad overview of this methodology can be seen in recent surveys [33, 19, 18, 29]. This tensor approach was first applied to parametric and stochastic PDEs in [26] based on the *canonical* format and it was further extended in [34, 32, 15] to the case of *Hierarchical Tucker*, as well as *Tensor Train* [41]

and *Quantized Tensor Train* formats. In this paper, we use the Quantized Tensor Train (QTT) format [38, 28, 39], which we describe in Section 2, and then show that the preconditioned system can be solved efficiently either by a direct method or by Krylov iterations demonstrating that these QTT-structured representations of the data appear to be highly efficient for the solution of the multiparametric problem.

We note that the algebraic construction using a direct solution formula in the present paper is motivated by the available results on low-rank approximate inverses of parametric elliptic operators [29] and on the structure of the reciprocal preconditioner [9]. It is due to the direct formula, available only in the case of one physical dimension, that the discretizations of parametric PDEs can be solved fully using only the basic algorithms of the TT-structured arithmetic. For higher-dimensional problems, efficient (but heuristic) tensor algorithms based on cross approximation are being developed [3, 10].

The sparse grid approach [37, 48, 6, 35], best N-term approximations [7] and adaptive Galerkin discretizations [14] could be used instead of tensor-structured representations to determine a reduced collocation set and, consequently, the complexity of the resulting problem. Further cost reduction in the sparse grid collocation or stochastic Galerkin methods can be achieved by a low-rank separation of physical and stochastic variables [15, 36, 5]. Another state-of-the-art technique for various high-dimensional problems, including stochastic PDEs, is the Monte Carlo and Quasi Monte Carlo methods [16, 17, 47].

We adhere to the collocation on full tensor-product grids and let the QTT format select the "effective" degrees of freedom adaptively. In the extreme case, when the *QTT ranks* are bounded by r uniformly in $\mathfrak{n}$ and M, the QTT format ensures *logarithmic* complexity $\mathcal{O}\left(\mathsf{M}\mathsf{r}^2 \log \mathfrak{n}\right)$. For parametric problems, although the TT ranks often exhibit the dependence $\mathsf{r} \sim \mathsf{M}$, the total storage $\mathcal{O}\left(\mathsf{M}^3\right)$ is still much smaller than $\mathfrak{n}^{\mathsf{M}}$. This makes the problem tractable for high M. The QTT representation may be used also for the physical variable $\mathsf{x} \in \mathsf{D}$.

The basics of the QTT-approximation theory for function-related tensors were given in [28, 28]. Since then, the QTT format has been successfully applied to such problems as the solution of linear systems, eigenvalue problems, and in the discrete Fourier analysis, see [27, 43, 31, 30, 25, 23, 12, 24]. There are also results on the analytical QTT structure of function-related vectors and discretized operators, see [22, 42, 21, 44].

We propose two ways to solve (1.1) in the QTT representation. First, Krylov solvers can be preconditioned by the approach introduced in [9] for problems of the form (1.2). This preconditioner is based on the reciprocal diffusion coefficient $1/\mathfrak{a}$ and, in the one-dimensional case, ensures that the preconditioned matrix of

the discrete problem has only two different eigenvalues. In the present paper we generalize this preconditioner to the multiparametric problem (1.1) and show how it can be used to precondition tensor-structured Krylov iterative solvers of (1.1). In particular, in our numerical experiments we consider the GMRES method.

Second, the clustering properties for the preconditioner allow to derive a direct solution formula for (1.2), and, henceforth, for (1.1) at each fixed y. We show how this formula can be evaluated for all y simultaneously with the use of QTT-structured vector operations, such as the pointwise multiplication and summation.

The paper is organized as follows. In Section 2 we present a basic description of the TT and QTT formats. In Section 3 we consider a weak formulation of (1.1), whose QTT-structured discretization follows in Section 4. In Section 5 we present the reciprocal preconditioner, generalize it to the parametric case, and show our main result, how a direct resolution formula can be derived. In Section 6 we outline technical aspects of the computation of the reciprocal coefficient, and calculation with the preconditioner and the direct formula. Finally, Section 7 presents numerical results obtained with the use of our approach.

2. The QTT format for the low-parametric data representation

As we outlined in the introduction, the discrete problem we proceed to from (1.1) suffers from the "curse of dimensionality". For example, the elementwise representation requires storing $O(Nn^M)$ entries of vectors of the diffusion coefficient, right-hand side and solution. To avoid this, we represent the vectors and matrices involved by their low-parametric decompositions in the Tensor Train (TT) and Quantized Tensor Train (QTT) formats. Below we give basic definitions and for further details refer to [41] and [28, 39].

Definition 2.1. *An* M*-dimensional* $n_1 \times \ldots \times n_M$ *tensor* $\mathbf{u}$ *is said to be represented in the TT format through parametric matrices* $u^{(k)}$, $k = 1, \ldots, d$, *which are called* TT cores*, if it holds that*

$$\mathbf{u}(i_1, \ldots, i_M) = u^{(1)}(i_1) \cdot \ldots \cdot u^{(M)}(i_M) \quad \textit{for} \quad i_k = 1, \ldots, n_k, \quad k = 1, \ldots, M,$$

where $u^{(k)}(i_k)$ *is a matrix of size* $r_{k-1} \times r_k$ *for* $1 \le i_k \le n_k$, $k = 1, \ldots, M$, *and* $r_0 = r_M = 1$. *The summation limits* r_k *are referred to as* TT ranks *of this particular decomposition of* $\mathbf{u}$.

The same data can be seen as an $n_1 \cdots n_M$-vector u, by the simple renumeration

$$u(\mathbf{i}) = \mathbf{u}(i_1, \ldots, i_M), \quad \mathbf{i} = \overline{i_1 \ldots i_M} = i_M + (i_{M-1} - 1)n_M + \cdots + (i_1 - 1)n_M \cdots n_2.$$

A matrix A of the same sizes as u can be represented by a similar structure. Let A be indexed by row indices $\overline{i_1 \dots i_M}$ and column indices $\overline{j_1 \dots j_M}$. For every $k = 1, \dots, M$ we merge i_k and j_k, which yields a M-dimensional $n_1^2 \times \dots \times n_M^2$ tensor $\mathbf{A}$ defined as

$$A(\overline{i_1 \dots i_M}, \overline{j_1 \dots j_M}) = \mathbf{A}(\overline{i_1 j_1}, \dots, \overline{i_M j_M}).$$

Then we apply the TT representation to $\mathbf{A}$ according to Definition 2.1 and, for $k = 1, \dots, M$, split back the indices i_k and j_k in the core, $A^{(k)}(i_k, j_k)$.

The TT format allows efficient linear algebra operations. For example, the product of an $n^M \times n^M$ matrix given in a TT decomposition of ranks $r_k \le r$, by an n^M-component vector given in a TT decomposition of ranks $r'_k \le r'$, is represented in the TT decomposition with TT ranks bounded by rr', and the complexity of this computation is linear with respect to the dimension M and polynomial with respect to the mode sizes $n = n_1 = \dots = n_M$ and ranks r, r'.

In order to introduce the Quantized Tensor Train (QTT) decomposition [28, 39], we assume that each of the indices i_k, $k = 1, \dots, M$, varies in the range from 1 to $n_k = 2^{l_k}$. Then it can be represented in the binary coding,

$$i_k = \overline{i_{k,1} \dots i_{k,l_k}} = i_{k,l_k} + 2(i_{k,l_k-1} - 1) + \dots + 2^{l_k-1}(i_{l_k,1} - 1), \quad i_{k,p} = 1, 2,$$

where $p = 1, \dots, l_k$. By such a "quantization", the M-dimensional $n_1 \times \dots \times n_M$ tensor $\mathbf{u}$ is transformed into a $(l_1 + \dots + l_M)$-dimensional $2 \times \dots \times 2$ tensor,

$$\mathbf{u}(i_{1,1} \dots i_{1,l_1}, \dots, i_{M,1} \dots i_{M,l_M}) := \mathbf{u}(i_1, \dots, i_M).$$

If this tensor is decomposed in the TT format according to Definition 2.1, this decomposition, its TT ranks and TT cores are called respectively *QTT decomposition*, *QTT ranks* and *QTT cores*. We emphasize that the same data is also refereed to as a $2^{l_1+\dots+l_M}$-vector u. In the following, we will write standard vector operations (e.g. Au), implying the corresponding TT operations if A and u are defined by TT formats.

For certain classes of functions such as polynomials, exponentials and trigonometric functions, exact and explicit QTT decompositions of low ranks are available, see, e.g. [28]. As a result, for function-related data, the smoothness leading to the rapid convergence of expansions in terms of such systems results in the efficiency of the TT or QTT approximation.

For the efficiency of computations with the data recast in the format, it is crucial that the format can be maintained through the course of computations. This is achieved owing to that, for the TT and QTT formats, the problem of low-rank

approximation is well posed and can be solved quasi-optimally with the use of standard matrix algorithms, such as SVD and QR [41, 41]. We denote this *truncation* operation by

$$v = \mathcal{T}_\varepsilon(u),$$

where u is an input vector given in a TT decomposition of ranks bounded from above by R, and v is its approximation returned in a TT decomposition with the accuracy ε and nearly minimal possible ranks. The complexity of $\mathcal{T}$ is $\mathcal{O}(Mnr^3)$, where M is the dimension and n is the mode size of the tensor $\mathbf{u}$. In particular, in the case of QTT approximation we have $n = 2$.

Our implementation is based on the TT Toolbox by Ivan Oseledets et al, which is publicly available at http://github.com/oseledets/TT-Toolbox. The toolbox provides subroutines for various linear algebra operations in the TT format and includes also TT-structured solvers of linear systems (e.g. TT-GMRES [11] and the alternating DMRG [43] and AMEn [13] methods).

3. Multiparametric elliptic equation

3.1. Formulation of the problem

Let us return to the multiparametric problem (1.1). For the weak formulation of the problem we follow [26] and consider the tensor-product Hilbert space $V = V_x \otimes V_y$, where $V_x = H_0^1(D)$ and $V_y = L_2(\boldsymbol{Y})$. We assume that $f \in H^{-1}(D) \otimes L_2(\boldsymbol{Y})$ and $a \in C(D \times \boldsymbol{Y})$, $a(\cdot, y) \in C^2(D)$ for all $y \in \boldsymbol{Y}$. Here, $L_2(D)$ and $H^1(D)$ are the standard Lebesgue and Sobolev spaces of square-integrable functions and of functions with square-integrable weak derivatives; $H_0^1(D)$ is the subspace of $H^1(D)$ whose elements vanish at the boundary of D. For every $y \in \boldsymbol{Y}$ we introduce the associated bilinear form $A(y) : V_x \times V_x \to \mathbb{R}$:

$$A[w, v](y) = \int_D a(x, y) \frac{dw(x)}{dx} \frac{dv(x)}{dx} dx \quad \forall w, v \in V_x$$

and consider the following weak formulation: find $u(x, y) \in V$ such that

$$A[u, v](y) = \int_D f(x, y) v(x) dx \quad \forall v \in V_x, \quad \forall y \in \boldsymbol{Y}. \tag{3.1}$$

3.2. The form of the coefficient and right-hand side

The Karhunen–Loève expansion is a very common way to proceed from a stochastic problem to a multiparametric equation, see [26] and references therein. The dimensionality M of the parametric domain is exactly the truncation parameter of the Karhunen–Loève expansion, which may need to be large (tens, hundreds or even more). In the present paper, we consider the following two forms of the coefficient a.

In the *additive case* we have

$$a(x,y) = a_0(x) + \sum_{m=1}^{M} a_m(x) y_m, \tag{3.2}$$

where $a_m \in L_\infty(D)$, $m = 1, \dots, M$.

Assumption 3.1. *Following [26], we assume that there exist* $a_{\min} > 0$ *and* $\gamma \in (0,1)$*, such that for all* $x \in D$ *it holds that* $a_{\min} \le a_0(x) < \infty$ *and* $\left| \sum_{m=1}^{M} a_m(x) \right| < \gamma\, a_{\min}$ *for all* $x \in D$.

Assumption 3.1 ensures the strong ellipticity of the bilinear form $A(y)$ uniformly in $\boldsymbol{Y}$. As a result, the problem (3.1) has a unique solution $u \in V$ for any $f \in H^{-1}(D) \otimes L_2(\boldsymbol{Y})$. Moreover, the solution is analytic w.r.t. the parameters [8], which justifies Galerkin and collocation discretization schemes in y.

We also consider the *log-additive model* [26] with

$$a(x,y) = e^{a_0(x)} \prod_{m=1}^{M} e^{a_m(x) y_m}. \tag{3.3}$$

Due to the boundedness of $\boldsymbol{Y}$, uniform upper bounds on a_m, $1 \le m \le M$, ensure that (3.1) has a unique solution $u \in V$ for any $f \in H^{-1}(D) \otimes L_2(\boldsymbol{Y})$.

Following [26, 48, 6], we assume below that in both cases the right-hand side f is independent of the parameters, i.e. $f \in H^{-1}(D)$.

4. Discretization of the multiparametric problem in the physical and parametric domains

We use the Galerkin FEM with N piecewise-linear hat functions $\{\varphi_i\}$ in the physical domain, constructed on a uniform grid with step size $h = 1/(N+1)$ and

nodes $x_i = hi$, $i = 1, \dots, N$. The entries of the stiffness matrix $\Gamma_y[a]$ at each parameter point y read

$$(\Gamma_y[a])_{i,i'} = (a(\cdot, y)\varphi_i', \varphi_{i'}')_{L_2(D)}, \quad i, i' = 1, \dots, N.$$

We use the midpoint quadrature rule. This quadrature of accuracy $\mathcal{O}\left(h^2\right)$ yields a tridiagonal matrix

$$\Gamma_y[a] = \frac{1}{h}\begin{bmatrix} a_1 + a_2 & -a_2 & & & & \\ -a_2 & a_2 + a_3 & -a_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -a_{N-1} & a_{N-1} + a_N & -a_N & \\ & & & -a_N & a_N + a_{N+1} \end{bmatrix}, \tag{4.1}$$

where $a_i = a(x_{i-1/2}, y)$, $i = 1, \dots, N$. However, for the QTT decomposition we need the number of unknowns to be a power of 2, so we extrapolate $a_{N+1} = a_N$.

In the parametric domain we discretize the problem by collocation on an n^M-point full-tensor-product grid [26, 48, 6], which we denote as $\{y_{j_1,\dots,j_M}\}_{j_1,\dots,j_M}$. The fully discrete problem reads

$$\boldsymbol{\Gamma}[a]u = f, \tag{4.2}$$

where $u, f \in \mathbb{R}^{Nn^M}$. The matrix $\boldsymbol{\Gamma}[a]$ is block-diagonal of size $Nn^M \times Nn^M$:

$$\boldsymbol{\Gamma}[\boldsymbol{a}] = \begin{bmatrix} \Gamma_{y_{11\dots1}} & & & & & \\ & \Gamma_{y_{21\dots1}} & & & & \\ & & \ddots & & & \\ & & & \Gamma_{y_j} & & \\ & & & & \ddots & \\ & & & & & \Gamma_{y_{nn\dots n}} \end{bmatrix}, \quad \Gamma_{y_j} \in \mathbb{R}^{N\times N},$$

where $j = \overline{j_1, \dots, j_M}$ is a multi-index in the discrete parametric space, $j_k = 1, \dots, n$, $k = 1, \dots, M$, and Γ_{y_j} is the stiffness matrix (4.1) at $y = y_j$ fixed.

The entries of the right-hand side vector read $f_{i,j} = (f(x), \varphi_i(x))_{L_2(D)}$.

Now let us discuss the low-rank representations of the matrix and the right-hand side. If we can separate the physical variable and the parameters in the diffusion coefficient, so that

$$a_{i,j} = a(x_{i-1/2}, y_j) = \sum_{\alpha=1}^{r} a_\alpha^{(x)}(i) \cdot a_\alpha^{(y)}(j), \qquad a_\alpha^{(x)} \in \mathbb{R}^N, \quad a_\alpha^{(y)} \in \mathbb{R}^{n^M}, \tag{4.3}$$

with a moderate separation rank r, then the matrix can be represented as

$$\boldsymbol{\Gamma}[\boldsymbol{a}] \;=\; \sum_{\alpha=1}^{r} \Gamma_y\left[a_\alpha^{(x)}\right] \otimes \operatorname{diag}\left(a_\alpha^{(y)}\right), \tag{4.4}$$

where $\Gamma_y\left[a_\alpha^{(x)}\right]$ is built from $a_\alpha^{(x)}$ according to (4.1), and $\operatorname{diag}\left(a_\alpha^{(y)}\right)$ is a diagonal matrix with the values $a_\alpha^{(y)}(j)$ stretched along the diagonal. Within the additive model (3.2), the separation rank r in (4.3) is bounded by the number of parameters M.

The right-hand side is written as $f = f^{(x)} \otimes e^{(y)}$, where $f^{(x)}$ is an N-vector with entries $(f(x), \varphi_i(x))_{L_2(D)}$, and $e^{(y)}$ is an n^M-vector of all ones.

We represent the indices in physical and parametric spaces in the binary coding to consider QTT decompositions of the vectors and matrices involved in the discrete problem:

$$j = \overline{j_{1,1}, \ldots, j_{1,l}, \ldots, j_{M,1}, \ldots, j_{M,l}}, \quad i = \overline{i_1, \ldots, i_L},$$

where $L = \log_2 N$, $l = \log_2 n$ and all $j_{k,p}, i_p \in \{1, 2\}$. Then the parts of the coefficient a in (4.3) can be represented in the QTT format as follows:

$$\begin{aligned} a_\alpha^{(x)}(i) &= \sum_{\gamma_1,\ldots,\gamma_{L-1}=1}^{\rho_1,\ldots,\rho_{L-1}} a_{\gamma_1}^{(x,1)}(i_1) a_{\gamma_1,\gamma_2}^{(x,2)}(i_2) \cdots a_{\gamma_{L-1},\alpha}^{(x,L)}(i_L), \\ a_\alpha^{(y)}(j) &= \sum_{\beta_1,\ldots,\beta_{Ml-1}=1}^{r_1,\ldots,r_{Ml-1}} a_{\alpha,\beta_1}^{(y,1,1)}(j_{1,1}) \cdots a_{\beta_{Ml-2},\beta_{Ml-1}}^{(y,M,l-1)}(j_{M,l-1}) a_{\beta_{Ml-1}}^{(y,M,l)}(j_{M,l}). \end{aligned} \tag{4.5}$$

In other words, we consider a two-level structure: first we separate the spatial variable and parameters, and then we introduce a finer structure in each of them. The index α is placed in the last QTT block of $a^{(x)}$ and the first QTT block of $a^{(y)}$ in order to maintain a consistent QTT format for the entire a (4.3).

The QTT cores of the matrix $\boldsymbol{\Gamma}[a]$ can be written similarly, recalling that the vector is turned to the diagonal matrix without changing the TT ranks. First, define the parameter-related QTT block of the matrix $\boldsymbol{\Gamma}[a]$:

$$\Gamma_{\beta_{q-1},\beta_q}^{(y,q)} = \begin{bmatrix} a_{\beta_{q-1},\beta_q}^{(y,q)}(1) & \\ & a_{\beta_{q-1},\beta_q}^{(y,q)}(2) \end{bmatrix}, \tag{4.6}$$

where we denote $a^{(y,q)} \equiv a^{(y,k,p)}$ from (4.5) for $k = 1, \ldots, M$, and $p = 1, \ldots, l$, with the multi-index $q = p + (k-1)l = 1, \ldots, Ml$.

Without the quantization of the spatial index i, the spatial TT block of $\boldsymbol{\Gamma}[a]$ appears immediately as $\Gamma_\alpha^{(x)} = \Gamma_y\left[a_\alpha^{(x)}\right]$. Therefore, the whole stiffness matrix of the multiparametric problem writes as the following TT format:

$$\boldsymbol{\Gamma}[a] = \sum_{\alpha,\beta_1,\ldots,\beta_{Ml-1}} \Gamma_y\left[a_\alpha^{(x)}\right] \otimes \left(\Gamma^{(y,1)}_{\alpha,\beta_1} \otimes \left(\bigotimes_{q=2}^{Ml-1} \Gamma^{(y,q)}_{\beta_{q-1},\beta_q} \right) \otimes \Gamma^{(y,Ml)}_{\beta_{Ml-1}} \right).$$

The spatial part can be brought into the QTT format by using the shift matrices and their explicit rank-2 QTT representation [21, 22]. Let us denote the upper shift matrix by $S \in \mathbb{R}^{N\times N}$, then it holds

$$\Gamma_y\left[a_\alpha^{(x)}\right] = S\mathrm{diag}(a_\alpha^{(x)}) + \mathrm{diag}(a_\alpha^{(x)} + Sa_\alpha^{(x)}) + \mathrm{diag}(a_\alpha^{(x)})S^\top, \quad S_{i,i'} = \begin{cases} 1, & i' = i+1, \\ 0, & \text{else.} \end{cases} \tag{4.7}$$

Each multiplication by S multiplies the QTT ranks by 2, while the construction of the diagonal matrix keeps the ranks unchanged. Assuming that the QTT ranks of $a^{(x)}$ are bounded by $r(a^{(x)})$, we can estimate $r\left(\Gamma^{(x)}\right) \le (2+(1+2)+2)r(a^{(x)}) = 7r(a^{(x)})$.

The considerations given above lead to the following bound for the QTT ranks of the stiffness matrix.

Theorem 4.1. *Consider the problem (3.1) discretized with the use of the Galerkin-collocation method. Suppose that the diffusion coefficient a is given in a QTT decomposition (4.5) with the following TT rank bounds: $\rho_p \le R_x$, for $p = 1,\ldots,L-1$, $\alpha \le r$, and $r_q \le R_y$, for $q = 1,\ldots,Ml-1$.*

Then the matrix $\boldsymbol{\Gamma}[a(y)]$ can be represented in the QTT format, the parameter-related QTT ranks are bounded by R_y, the space-parameters separation rank is r, and the space-related QTT ranks are bounded by $7R_x$.

5. Iterative and direct solution of the parametric elliptic equation

5.1. Preconditioning of the multiparametric problem

Let us again start the consideration from a fixed point y. For the unitary coefficient, the matrix $\Gamma_y[a]$ coincides with the discretized Laplace operator: $\Gamma[1] = \Delta_h$. In [9] it was shown that

$$P_2 = \Delta_h^{-1}\Gamma_y[1/a]\Delta_h^{-1} \tag{5.1}$$

is an efficient preconditioner for (1.2). The preconditioning relies on the multiplication by the matrix $\Gamma_y[1/a]$ and the solution of two Laplace equations. The main properties of P_2 are as follows.

Theorem 5.1 ([9], Thm 3.1). *The preconditioner* P_2 *is spectrally equivalent to the inverse matrix* $\Gamma_{\mathsf{y}}[\mathfrak{a}]^{-1}$ *with constants independent of* $\mathfrak{h}$*:*

$$\frac{\min(\mathfrak{a})}{\max(\mathfrak{a})}\Gamma_{\mathsf{y}}[\mathfrak{a}]^{-1} \le \mathsf{P}_2 \le \frac{\max(\mathfrak{a})}{\min(\mathfrak{a})}\Gamma_{\mathsf{y}}[\mathfrak{a}]^{-1}.$$

In the presence of parameters, the preconditioner (5.1) generalizes to

$$\boldsymbol{P_2} = \boldsymbol{\Delta}_{\mathfrak{h}}^{-1}\boldsymbol{\Gamma}[1/\mathfrak{a}]\boldsymbol{\Delta}_{\mathfrak{h}}^{-1}, \tag{5.2}$$

where $\boldsymbol{\Delta}_{\mathfrak{h}} = \boldsymbol{\Gamma}[1] = \Delta_{\mathfrak{h}} \otimes \underbrace{\mathrm{I} \otimes \cdots \otimes \mathrm{I}}_{\mathsf{M}}$. Note that $\boldsymbol{\Delta}_{\mathfrak{h}}^{-1} = \Delta_{\mathfrak{h}}^{-1} \otimes \mathrm{I} \otimes \cdots \otimes \mathrm{I}$ is explicitly representable in the QTT format with the TT ranks at most 5 [22]. The tensor-structured computation of $\boldsymbol{\Gamma}[1/\mathfrak{a}]$ was explained in Section 4. In Section 6.1 we present QTT-structured Newton iterations, which allow to compute the QTT decomposition of the reciprocal coefficient $1/\mathfrak{a}$.

The considerations given above lead to the following QTT rank estimates.

Theorem 5.2. *Suppose that the reciprocal coefficient* $\mathfrak{b} = 1/\mathfrak{a}$ *is given in a QTT decomposition similar to* (4.5) *with the following TT rank bounds:* $\rho_{\mathsf{p}} \le \mathsf{R}_{\mathsf{x}}^{\mathrm{recp}}$*, for* $\mathsf{p} = 1, \dots, \mathsf{L}-1$*,* $\alpha \le \mathsf{r}^{\mathrm{recp}}$*, and* $\mathsf{r}_{\mathsf{q}} \le \mathsf{R}_{\mathsf{y}}^{\mathrm{recp}}$*, for* $\mathsf{q} = 1, \dots, \mathsf{M}\mathfrak{l}-1$*.*

Then the preconditioner $\boldsymbol{P_2}$ *can be represented in the QTT format, the parameter-related QTT ranks are bounded by* $\mathsf{R}_{\mathsf{y}}^{\mathrm{recp}}$*, the space-parameters separation rank is* $\mathsf{r}^{\mathrm{recp}}$*, and the space-related QTT ranks are bounded by* $175\mathsf{R}_{\mathsf{x}}^{\mathrm{recp}}$*.*

Remark 5.3. *The constant* 175 *appears from the multiplication* $5 \cdot 7 \cdot 5$ *of the QTT ranks of the first* $\Delta_{\mathfrak{h}}^{-1}$*, the stiffness matrix* $\boldsymbol{\Gamma}[1/\mathfrak{a}]$ *and the second* $\Delta_{\mathfrak{h}}^{-1}$*, respectively. In practice, the preconditioner* (5.2) *need not be assembled explicitly: it can be applied to a vector in the factorized form as follows:*

$$\mathcal{T}_\varepsilon(\boldsymbol{P_2}\mathcal{v}) = \mathcal{T}_\varepsilon\left(\boldsymbol{\Delta}_{\mathfrak{h}}^{-1}\ \mathcal{T}_\varepsilon\left(\boldsymbol{\Gamma}[1/\mathfrak{a}]\ \mathcal{T}_\varepsilon(\boldsymbol{\Delta}_{\mathfrak{h}}^{-1}\mathcal{v})\right)\right).$$

Since usually each of these truncations returns the ranks to average values, the real cost of $\boldsymbol{P_2}\mathcal{v}$ *is comparable with that of* $\boldsymbol{\Gamma}[\mathfrak{a}]\mathcal{v}$*.*

5.2. Direct resolution formula

General theorem 5.1 is applicable in the case of any domain D, and also in the parametric case, if we consider the maximum and minimum of $\mathfrak{a}$ w.r.t. both $\mathfrak{x}$ and $\mathfrak{y}$. However, for the one-dimensional D, the preconditioner (5.1) possesses a much stronger *clustering* property.

Theorem 5.4 ([9], Thm 2.2). *Assume* $D = (0,1)$, *then* $P_2\Gamma_y[a] = I + R$, *where* $R = \psi\eta^T$ *is a rank-1 matrix, and the elements of the vectors* $\psi, \eta \in \mathbb{R}^N$ *are as follows:*

$$\begin{aligned} \psi_i &= z_i \sum_{s=1}^{N} z_s \left(\tfrac{1}{a_{s+1}} - \tfrac{1}{a_s}\right) - \sum_{s=1}^{i-1} z_s \left(\tfrac{1}{a_{s+1}} - \tfrac{1}{a_s}\right) + z_i \left(\tfrac{1}{a_i} - \tfrac{1}{a_N}\right), \\ \eta_i &= a_{i+1} - a_i, \\ z_i &= ih = i/(N+1), \end{aligned}$$

where $i = 1, \ldots, N$, *and all coefficients* a_i *are taken at the given point* y *(which we omit for brevity).*

Let us write the inverse of the stiffness matrix $\Gamma_y[a]$ explicitly with the use of this representation. Consider the preconditioned system $P_2\Gamma_y[a]u = P_2 f$, where $P_2\Gamma_y[a] = I + \psi\eta^T$. The solution is $u = \left(I + \psi\eta^T\right)^{-1} P_2 f$. By the Sherman-Morrison formula [45, 4], we obtain

$$\left(I + \psi\eta^T\right)^{-1} = I - \frac{\psi\eta^T}{1 + \eta^T\psi}.$$

This yields an explicit formula for the solution in terms of the reciprocal coefficient and the matrix-vector multiplication by Δ_h^{-1}:

$$u_y = \Gamma_y[a]^{-1} f = \left(I - \frac{\psi\eta^T}{1 + \eta^T\psi}\right) \Delta_h^{-1}\Gamma_y[1/a]\Delta_h^{-1} f, \tag{5.3}$$

which needs to be evaluated for all $y \in \boldsymbol{Y}$, and ψ, η are given in (5.3).

Remark 5.5. *For simplicity, throughout the paper, we consider the Dirichlet boundary conditions. However, analogs of* (5.3) *can be derived in the case of Robin boundary conditions* $\left(\alpha u + \beta \frac{\partial u}{\partial n}\right)|_{\partial D} = 0$ *as well.*

6. Implementation details

6.1. Tensor-structured computation of the reciprocal diffusion coefficient

The preconditioner (5.1) and the direct solution formula (5.3) involve the reciprocal diffusion coefficient $1/a$, whose computation by elementwise representation does not preserve the tensor-structured form. We employ the Newton method, which is widely used for the iterative inversion of matrices [20] and can be easily adapted to tensor-structured formats.

To compute $1/a$ in the TT and QTT formats, we need to multiply vectors elementwise and truncate them in these formats. We perform the s-th Newton iteration for the computation of $b_s \approx 1/a$ as follows:

$$\begin{aligned} c_s &= \mathcal{T}_\varepsilon(b_s a b_s), \\ b_{s+1} &= \mathcal{T}_\varepsilon(2b_s - c_s), \end{aligned}$$

where ab denotes the Hadamard product of vectors $a, b \in \mathbb{R}^{Nn^M}$. The method converges quadratically, provided that $|1 - ab_0| \le 1$ for all entries. For fast computation of the Hadamard product in the TT format, we note that ab can be written as a matrix-vector product $\operatorname{diag}(a)b$. The matrix $\operatorname{diag}(a)$ is representable in the TT format with the same TT ranks as a, and we may use the DMRG algorithm for the fast approximation of the matrix-vector product [40].

In the log-additive case (3.3) the reciprocal coefficient is computed as in [32]. The method is based on the low-rank QTT structure of a at a single spatial point. It is formulated as follows:

1: Set $a = 0$.
2: **for** $i = 1, \dots, N$ **do**
3: compute the QTT format of $a_i = \left[a(x_{i-1/2}, y_j)\right]_{j=1}^{n^M}$,
4: add it to the representation: $a = a + I_i \otimes a_i$,
5: perform the compression $a = \mathcal{T}_\varepsilon(a)$.
6: **end for**

Here, $I_i \in \mathbb{R}^N$ is the i-th column of the identity matrix. Each term can be represented as a rank-1 QTT tensor, since

$$a(x_{i-1/2}, y_j) = \exp(a_0(x_{i-1/2})) \prod_{m=1}^{M} \exp\left(a_m(x_{i-1/2}) y_m(j_m)\right).$$

see also [28, 42]. Therefore the QTT ranks of a are bounded by N. The reciprocal coefficient in the log-additive case is computed with the same complexity through the summation of the terms $\exp(-a(x_{i-1/2}, y_j))$.

6.2. Preconditioned iterative solver

The spectral properties of preconditioner (5.1) (see Theorem 5.1) ensure the efficiency of Krylov iterative solvers for the solution of the preconditioned problem (4.2). Moreover, the theory of inexact Krylov methods [46] controls the convergence with respect to the error introduced by using low-rank approximations of a and $1/a$ and allows to increase the approximation tolerance ε at later iterations [11]. However, this strategy may lead to the stagnation of the residual at the level $\mathcal{O}(\operatorname{cond}(PA)\varepsilon)$.

6.3. Calculations via direct solution formula

Let us show how formula (5.3) should be evaluated in a low-rank tensor format. Assume that a and $b = 1/a$ are given in a separable form:

$$a_{i,j} = \sum_{\alpha=1}^{r_a} a_\alpha^{(x)}(i) \cdot a_\alpha^{(y)}(j), \quad b_{i,j} = \sum_{\zeta=1}^{r_b} b_\zeta^{(x)}(i) \cdot b_\zeta^{(y)}(j).$$

The vectors ψ, η in (5.3) are defined at every y_j independently, and for all $\alpha = 1, \dots, r_a$, $\zeta = 1, \dots, r_b$, and $i = 1, \dots, N$ we may compute

$$\begin{aligned} \delta a_\alpha^{(x)}(i) &= a_\alpha^{(x)}(i+1) - a_\alpha^{(x)}(i), \\ \delta b_\zeta^{(x)}(i) &= b_\zeta^{(x)}(i+1) - b_\zeta^{(x)}(i), \\ \psi_\zeta^{(x)}(i) &= z_i \sum_{s=1}^{N} z_s \delta b_\zeta^{(x)}(s) - \sum_{s=1}^{i-1} z_s \delta b_\zeta^{(x)}(s) + z_i \left(b_\zeta^{(x)}(i) - b_\zeta^{(x)}(N) \right), \\ \eta_\alpha^{(x)}(i) &= \delta a_\alpha^{(x)}(i); \end{aligned} \tag{6.1}$$

and then, assemble

$$\psi_i(j) = \sum_{\zeta=1}^{r_b} \psi_\zeta^{(x)}(i) b_\zeta^{(y)}(j), \qquad \eta_i(j) = \sum_{\alpha=1}^{r_a} \eta_\alpha^{(x)}(i) a_\alpha^{(y)}(j). \tag{6.2}$$

If $a^{(x)}$ and $b^{(x)}$ are kept in the full format, the computation is straightforward. In the QTT format, we may use the shift matrix (4.7) to compute

$$\delta a_\alpha^{(x)} = (S - I) a_\alpha^{(x)}, \qquad \delta b_\alpha^{(x)} = (S - I) b_\alpha^{(x)}.$$

The polynomial vector $[z_i] = [i/(N+1)]$ is also exactly representable in the rank-2 QTT format [28, 42]. The first sum in $\psi^{(x)}$ in (6.1) is the scalar product of z and $\delta b^{(x)}$, and the second sum can be efficiently computed as a multiplication of a triangular Toeplitz matrix by a vector [21]. The most time-consuming step is therefore the computation of a and $1/a$ themselves, which is, however, a precomputing step.

Let us introduce the vector $v = \boldsymbol{P_2} f$, indexed similarly to (6.2):

$$v = \boldsymbol{\Delta}_h^{-1} \boldsymbol{\Gamma}[1/a(y)] \boldsymbol{\Delta}_h^{-1} f = [v_i(j)]_{i,j=1}^{N,n^M}.$$

The solution formula (5.3) for all $y \in \boldsymbol{Y}$ can be now computed as follows:

$$u = \boldsymbol{\Gamma}[a]^{-1} f = v - \psi \cdot \left(e^{(x)} \otimes \frac{1}{e^{(y)} + \sum_{i=1}^{N} \eta_i \cdot \psi_i} \right) \cdot \left(e^{(x)} \otimes \left(\sum_{i=1}^{N} \eta_i \cdot v_i \right) \right), \tag{6.3}$$

where $e^{(x)}$ and $e^{(y)}$ are vectors of all ones of sizes N and n^M, respectively, and the Hadamard products ("$\cdot$") are computed in the QTT format, which consumes the most amount of time. We also need to perform the pointwise inversion of the denominator $e^{(y)} + \sum_{i=1}^{N} \eta_i \cdot \psi_i$. This can be done efficiently by the Newton iterations described in Section 6.1, as the denominator entries are usually very close to 1 (up to $\mathcal{O}(h)$).

6.4. Complexity

Given the maximal rank bounds R for the coefficient a, R^{recp} for $1/a$, R^f for the right-hand side, and taking into account the complexities of TT operations, we may conclude that the multiplications $\boldsymbol{\Gamma}[a]f$ and $\boldsymbol{\Gamma}[1/a]f$ in the QTT format can be evaluated with $\mathcal{O}\left((Ml + L)(R^f R)^2\right)$ and $\mathcal{O}\left((Ml + L)(R^f R^{\mathrm{recp}})^2\right)$ costs, respectively. However, the TT approximation of these products $\mathcal{T}_\varepsilon(\boldsymbol{\Gamma}[1/a]f)$ has the cubic complexity in the rank, $\mathcal{O}\left((Ml + L)(R^f R^{\mathrm{recp}})^3\right)$. Recalling Remark 5.3, we can conclude that the latter estimate also applies to the preconditioning operation $\boldsymbol{P_2}f$.

For Formula (6.3) we note that the rank of ψ is bounded by R^{recp}, the rank of η by R, and the rank bound of ν appears as the product $R^{\mathrm{recp}}R^f$. Therefore, the rightmost brackets in (6.3) require $\mathcal{O}\left((Ml + L)(R^{\mathrm{recp}}R^f R)^2\right)$ operations. The denominator can be constructed with $\mathcal{O}\left((Ml + L)(R^{\mathrm{recp}}R)^2\right)$ cost, but we cannot generally predict the rank of the Newton approximations in each iteration. Nevertheless, since the denominator is close to 1, the rank of the final reciprocal denominator is usually of the order of 10 and can be neglected. Finally, the multiplication with the leftmost ψ in (6.3) introduces another R^{recp} to the worst case estimate. In practice, we can make approximations after each operation (see Remark 5.3) and reduce the complexity.

7. Numerical experiments

In our numerical experiments we consider the additive and log-additive coefficient a, introduced in 3.2, with the polynomial and exponential decay of the Karhunen–Loève expansion:

- (polynomial) $a_m(x) = \dfrac{1}{2}\dfrac{1}{(m+1)^2}\sin(\pi m x)$, $m = 1, \dots, M$,
- (exponential) $a_m(x) = \exp(-0.7m)\sin(\pi m x)$, $m = 1, \dots, M$.

The mean field is always constant, $a_0 = 1$. The tensor rounding accuracy is set equal to $\varepsilon = 10^{-5}$, and the vector of ones, $f = (1, \dots, 1)^\top$, is taken for the right-hand side.

7.1. Experiments with the direct solution formula

In this series of examples, we present the time of computing the reciprocal coefficient, of multiplying the right-hand side by $\boldsymbol{P_2}$ and of performing the Sherman-Morrison correction. For an error indicator, we use the relative residual of the preconditioned system, $\|\boldsymbol{P_2}(\boldsymbol{\Gamma}\mathsf{u}-\mathsf{f})\|/\|\boldsymbol{P_2}\mathsf{f}\|$, since the preconditioner $\boldsymbol{P_2}$ is spectrally equivalent to $\boldsymbol{\Gamma}^{-1}$, and the preconditioned residual provides relevant information on the error in the solution.

7.1.1. Additive case, polynomial decay (Figures 7.1-7.4, Table 7.1)

In the additive case, the coefficient a is computed in seconds due to explicit canonical format with the rank bounded by $\mathsf{M}+1$. However, the pointwise coefficient inversion is more time-consuming. Here we exploit the Newton iteration, stopped by the criterion $\|\mathsf{b}_{s+1}-\mathsf{b}_s\| < \varepsilon\|\mathsf{b}_{s+1}\|$. The CPU times of this process are shown in Figures 7.1 and 7.2.

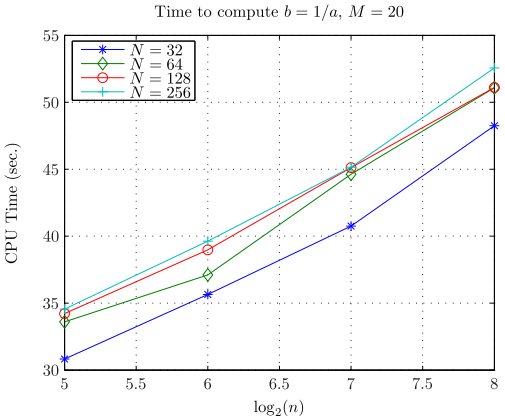

Figure 7.1: Time (sec.) of computing $1/\mathsf{a}$ versus N, n. $\mathsf{M} = 20$. Additive case, polynomial decay.

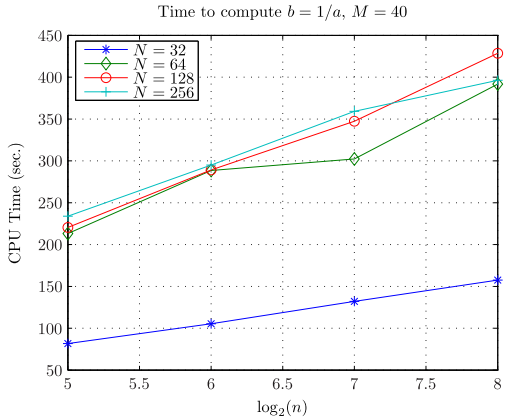

Figure 7.2: Time (sec.) of computing $1/\mathsf{a}$ versus N, n. $\mathsf{M} = 40$. Additive case, polynomial decay.

The times of the multiplication $\mathsf{v} = \boldsymbol{P_2}\mathsf{f}$ are not greater than 4 seconds (for the worst case $\mathsf{N} = \mathsf{n} = 256$, $\mathsf{M} = 40$, corresponding to $\mathsf{N}\mathsf{n}^{\mathsf{M}} \sim 10^{98}$ degrees of freedom in total), and are negligible with respect to the other solution steps.

The last step is the Sherman-Morrison correction (Figures 7.3, 7.4). The times are large due to large QTT ranks (of about 70) of the Hadamard products.

In Table 7.1 we present the preconditioned residual corresponding to the solution obtained by the explicit formula. We note that the residual stays at the level of 5ε independently on n, N and M.

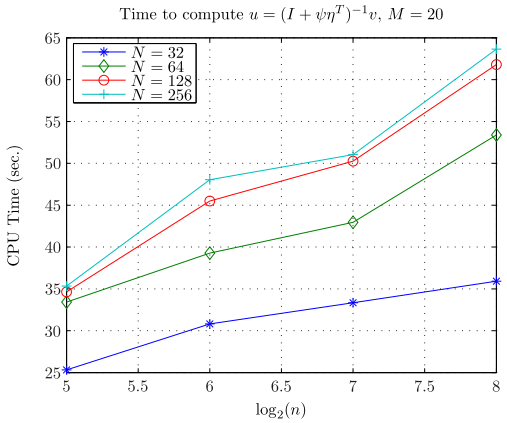

Figure 7.3: Time (sec.) of computing $\mathfrak{u} = (\mathrm{I}+\psi\eta^{\mathsf{T}})^{-1}\nu$ versus N, $\mathfrak{n}$. $\mathsf{M} = 20$. Additive case, polynomial decay.

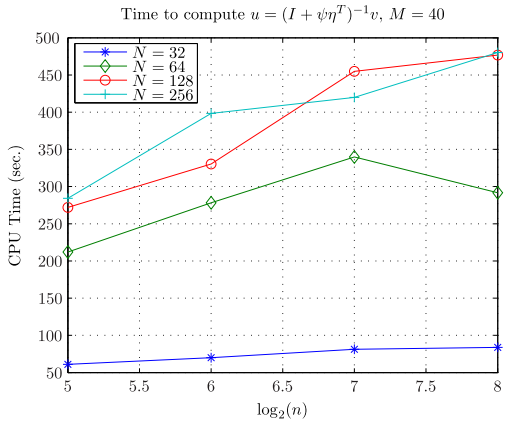

Figure 7.4: Time (sec.) of computing $\mathfrak{u} = (\mathrm{I}+\psi\eta^{\mathsf{T}})^{-1}\nu$ versus N, $\mathfrak{n}$. $\mathsf{M} = 40$. Additive case, polynomial decay.

Table 7.1: Preconditioned residuals $\dfrac{\|\boldsymbol{P}_2(\boldsymbol{\Gamma}\mathfrak{u} - \mathfrak{f})\|}{\|\boldsymbol{P}_2\mathfrak{f}\|}$ versus N, $\mathfrak{n}$ and M. Additive case, polynomial decay.

M	20				40			
N \ n	32	64	128	256	32	64	128	256
32	4.8e-5	4.9e-5	4.8e-5	4.9e-5	4.3e-5	5.4e-5	4.5e-5	6.9e-5
64	2.5e-5	4.9e-5	4.8e-5	3.1e-5	4.0e-5	3.3e-5	3.6e-5	4.0e-5
128	3.3e-5	3.8e-5	5.2e-5	5.1e-5	4.2e-5	4.3e-5	4.4e-5	4.2e-5
256	4.2e-5	4.2e-5	5.0e-5	4.7e-5	4.1e-5	4.5e-5	3.9e-5	4.9e-5

7.1.2. Log-additive case, polynomial decay (Figures 7.5-7.8)

To compute both $\mathfrak{a}$ and $1/\mathfrak{a}$ in this example, we use the add-and-compress procedure over the $\mathfrak{x}$-index, described in Section 6.1. As the ranks of $\mathfrak{a}$ and $1/\mathfrak{a}$ are approximately the same, so is the time needed to compute them. In Figures 7.5 and 7.6 we present the time of computing $1/\mathfrak{a}$ only.

Compared to the previous example, the complexity is almost cubic (not logarithmic) with respect to N. The reason is that within this procedure we compress tensors with ranks of order N.

The time needed to compute the Sherman-Morrison correction is also large (see Figure 7.7, Figure 7.8) due to the slow decay of the expansion (3.3) that results in large TT ranks.

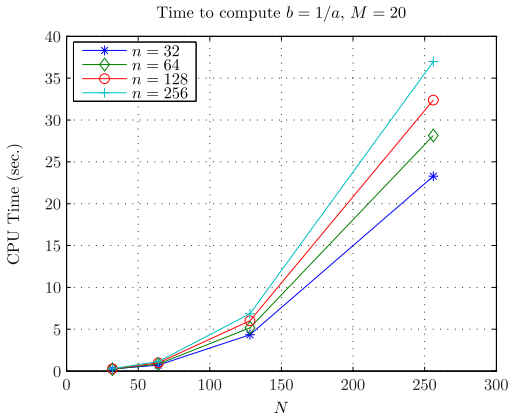

Figure 7.5: Time (sec.) of computing $1/a$ versus N, n. $M = 20$. Log-additive case, polynomial decay.

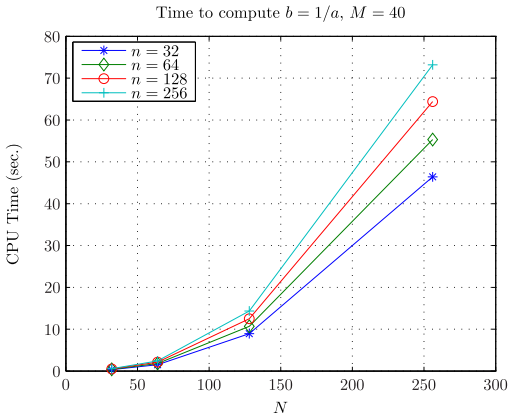

Figure 7.6: Time (sec.) of computing $1/a$ versus N, n. $M = 40$. Log-additive case, polynomial decay.

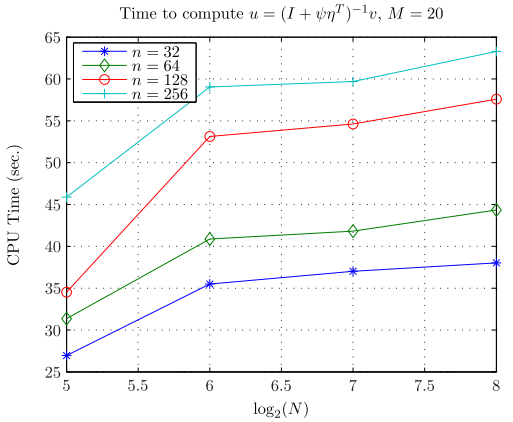

Figure 7.7: Time (sec.) of computing $u = (I + \psi\eta^T)^{-1}v$ versus N, n. $M = 20$. Log-additive case, polynomial decay.

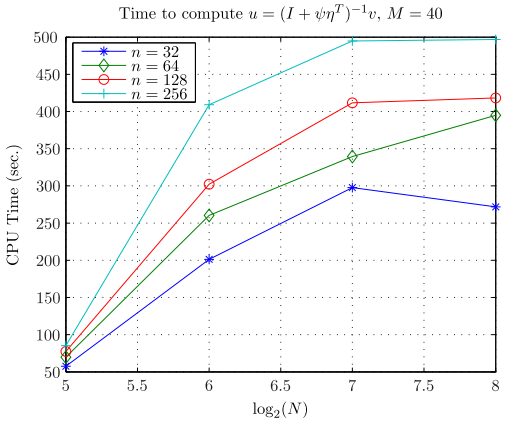

Figure 7.8: Time (sec.) of computing $u = (I + \psi\eta^T)^{-1}v$ versus N, n. $M = 40$. Log-additive case, polynomial decay.

7.1.3. Additive case, exponential decay (Figures 7.9-7.10)

The exponential decay in (3.2) ensures lower QTT ranks of a, and the Newton iterations for the computation of $1/a$ and the Sherman-Morrison correction are much faster than for the polynomial decay, see Figures 7.9 and 7.10.

As explained in Section 7.2.2, the (relatively low) exponential rate 0.7 yields even larger variance of the coefficient with respect to y, than in the polynomial case. Thus, the condition number of the stiffness matrix is larger, and the Krylov solver requires more iterations to converge. This leads to a larger computation time consumed

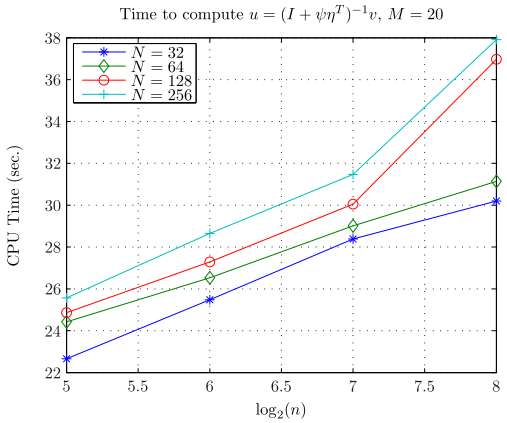

Figure 7.9: Time (sec.) of computing $\mathfrak{u} = (\mathrm{I}+\psi\eta^{\mathsf{T}})^{-1}\nu$ versus N, $\mathfrak{n}$. $\mathsf{M} = 20$. Additive case, exponential decay.

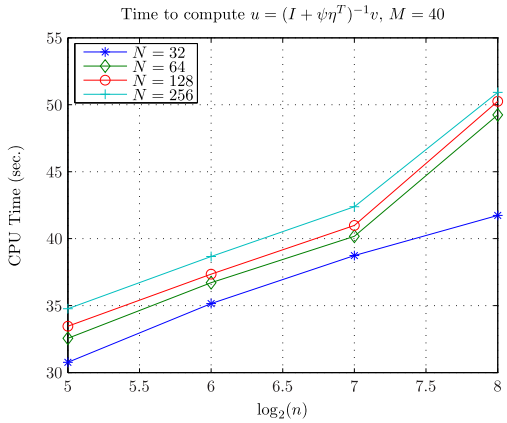

Figure 7.10: Time (sec.) of computing $\mathfrak{u} = (\mathrm{I}+\psi\eta^{\mathsf{T}})^{-1}\nu$ versus N, $\mathfrak{n}$. $\mathsf{M} = 40$. Additive case, exponential decay.

by the GMRES method (see Figures 7.17 and 7.18), while the Sherman-Morrison approach does not depend on the condition number and performs remarkably faster.

7.1.4. Log-additive case, exponential decay (Figures 7.11-7.14)

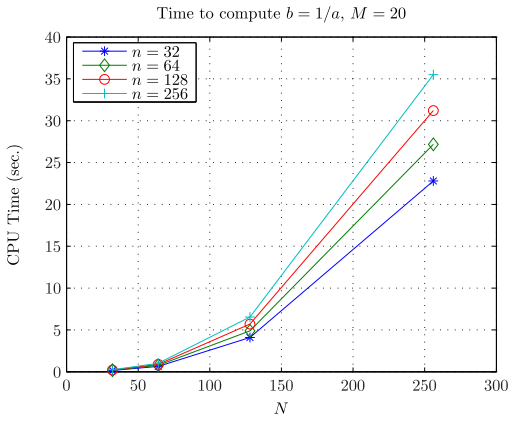

Figure 7.11: Time (sec.) of computing $1/\mathfrak{a}$ versus N, $\mathfrak{n}$. $\mathsf{M} = 20$. Log-additive case, exponential decay.

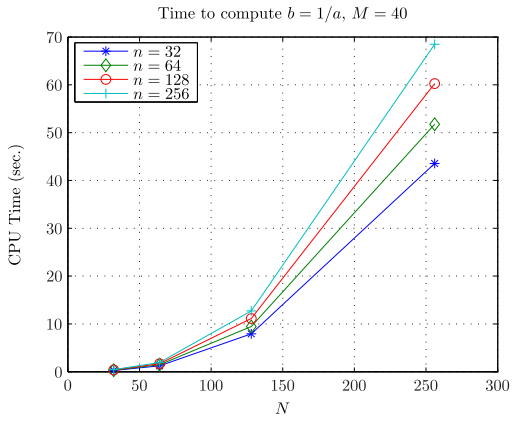

Figure 7.12: Time (sec.) of computing $1/\mathfrak{a}$ versus N, $\mathfrak{n}$. $\mathsf{M} = 40$. Log-additive case, exponential decay.

Again, in the log-additive case the times to compute $\mathfrak{a}$ and $1/\mathfrak{a}$ are approximately the same. As in Section 7.1.2, we observe a sublinear complexity of the add-and-compress technique (Fig. 7.11 and Fig. 7.12). The log-additive coefficient

model also leads to a faster rank growth in M: for $M = 20$, the times are smaller than in the additive case (see $N = 256$ and $n = 128, 256$), while for $M = 40$ the complexity becomes larger. However, the computation time for the Sherman-Morrison correction is the smallest in this series of experiments (Fig. 7.13,7.14).

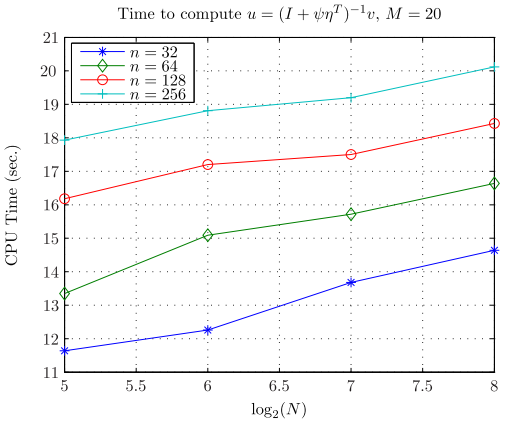

Figure 7.13: Time (sec.) of computing $\mathbf{u} = (\mathrm{I} + \psi\eta^{\mathsf{T}})^{-1}\nu$ versus N, n. $M = 20$. Log-additive case, exponential decay.

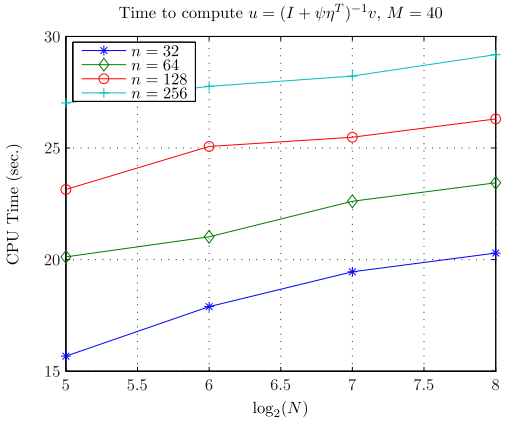

Figure 7.14: Time (sec.) of computing $\mathbf{u} = (\mathrm{I} + \psi\eta^{\mathsf{T}})^{-1}\nu$ versus N, n. $M = 40$. Log-additive case, exponential decay.

7.2. Preconditioned TT-GMRES

In this section we consider the TT-GMRES method [11], preconditioned with $\boldsymbol{P_2}$. We track the computation time of the TT-GMRES and the residual of the preconditioned problem.

7.2.1. Additive case, polynomial decay (Figures 7.15-7.16, Table 7.2)

First, we present the computation time of the GMRES solver. For $M = 40$, the GMRES is generally faster than the Sherman-Morrison correction. This is because the variations in the coefficient are small, the condition number is low and the convergence of the GMRES solver is fast. Contrarily, the Sherman-Morrison approach faces Hadamard products of vectors with large TT ranks.

In all tests the TT-GMRES converged in 3 iterations with the computed residual below ε. However, the actual residual might be larger, up to $\mathcal{O}(\mathrm{cond}(\mathrm{P}_2\boldsymbol{\Gamma})\varepsilon)$. This can be seen in Table 7.2: the residuals differ by an order of magnitude for different parameters. Recall that the direct solver is free from this effect (Table 7.1).

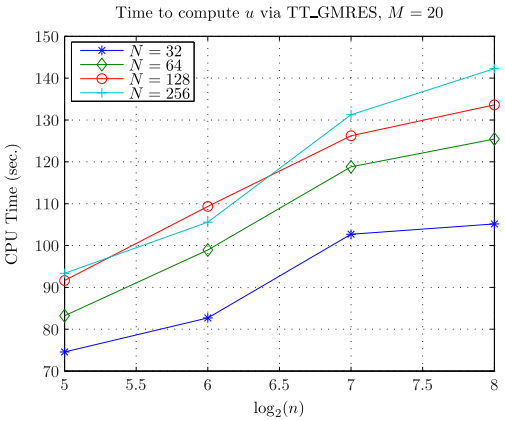

Figure 7.15: Time (sec.) of the TT-GMRES versus $\mathfrak{n}$. $\mathsf{M} = 20$. Additive case, polynomial decay.

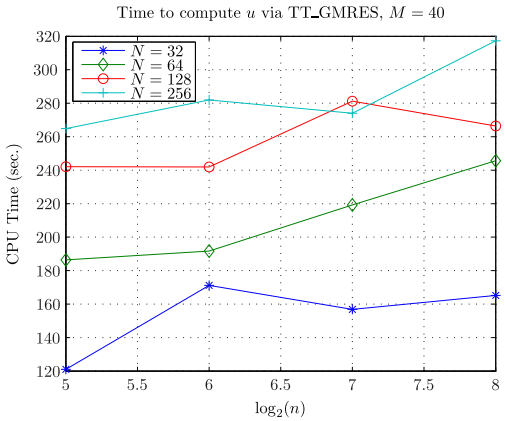

Figure 7.16: Time (sec.) of the TT-GMRES versus $\mathfrak{n}$. $\mathsf{M} = 40$. Additive case, polynomial decay.

Table 7.2: Residual $\dfrac{\|\boldsymbol{P_2}(\boldsymbol{\Gamma}\mathfrak{u} - \mathfrak{f})\|}{\|\boldsymbol{P_2}\mathfrak{f}\|}$ of the preconditioned problem versus N, $\mathfrak{n}$ and M. Additive case, polynomial decay.

M	20				40			
N \ n	32	64	128	256	32	64	128	256
32	2.3e-6	2.4e-6	2.5e-6	2.6e-6	1.8e-5	2.0e-5	2.4e-5	2.2e-5
64	2.8e-6	3.0e-6	2.7e-6	3.7e-6	1.9e-5	1.7e-5	2.1e-5	2.2e-5
128	3.0e-6	3.0e-6	3.3e-6	4.9e-6	1.9e-5	1.9e-5	2.0e-5	1.9e-5
256	3.2e-6	3.3e-6	3.0e-6	4.2e-6	1.9e-5	1.9e-5	1.8e-5	2.3e-5

7.2.2. Additive case, exponential decay (Figures 7.17-7.18)

For the exponential decay we observe a different picture. Despite the exponential decay of the coefficient, the quotient $\max(\mathfrak{a})/\min(\mathfrak{a})$ is greater than in the polynomial case. So is the condition number, and hence the number of GMRES iterations grows. This increases the computation times by one order of magnitude compared to the Sherman-Morrison correction (Figures 7.9 and 7.10). It shows the obvious advantages of the direct formula in the cases when coefficient $\mathfrak{a}$ varies significantly, but has relatively low TT ranks.

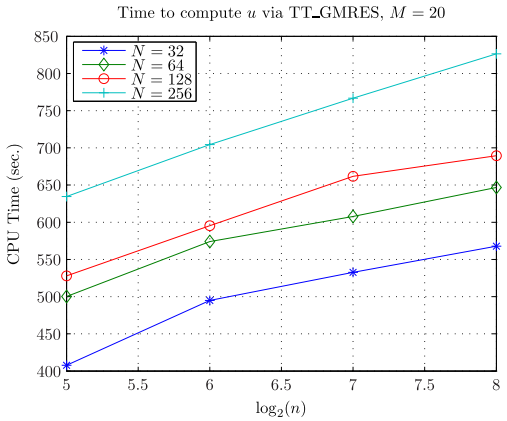

Figure 7.17: Time (sec.) of the TT-GMRES versus N, n. M = 20. Additive case, exponential decay.

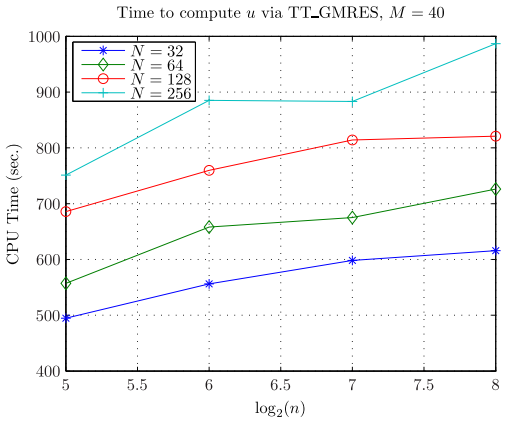

Figure 7.18: Time (sec.) of the TT-GMRES versus N, n. M = 40. Additive case, exponential decay.

8. Conclusion

We introduced a tensor-structured preconditioned iteration scheme and a direct solution method for a discretized one-dimensional second-order elliptic problem with a high-dimensional parameter. The use of the TT and QTT formats for the data-sparse representation within the GMRES solver and the explicit solution formula based on the Sherman-Morrison inversion, applied to the preconditioned equation, was shown to circumvent the "curse of dimensionality" and solve the problem efficiently. In our numerical experiments we compared the GMRES solver and the Sherman-Morrison inversion and found the latter competitive when the diffusion coefficient varies significantly and the conditioning of the problem is worse. The direct formula is only applicable in the case of a single physical dimension. The development of efficient tensor-structured algorithms for stochastic PDEs in more general settings is a topic of ongoing research [2, 3, 10].

References

[1] I. Babuska, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM J. Numer. Anal. 45 (2007) 1005–1034.

[2] J. Ballani, L. Grasedyck, Hierarchical tensor approximation of output quantities of parameter-dependent PDEs, SIAM J. Uncertainty Quantification 3 (2015) 852–872.

[3] J. Ballani, D. Kressner, Reduced basis methods: From low-rank matrices to low-rank tensors, SIAM J. Sci. Comput. 38 (2016) A2045–A2067.

[4] M.S. Bartlett, An inverse matrix adjustment arising in discriminant analysis, Ann. Math. Stat. 22 (1951) pp. 107–111.

[5] P. Benner, A. Onwunta, M. Stoll, Low-rank solution of unsteady diffusion equations with stochastic coefficients, SIAM/ASA Journal on Uncertainty Quantification 3 (2015) 622–649.

[6] M. Bieri, C. Schwab, Sparse high order FEM for elliptic sPDEs, Comput. Methods Appl. Mech. Eng. 198 (2009) 1149–1170.

[7] A. Cohen, R. DeVore, C. Schwab, Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs, Found. Comput. Math. 10 (2010) 615–646.

[8] A. Cohen, R. DeVore, C. Schwab, Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs, Anal. and Appl. 09 (2011) 11–47.

[9] S. Dolgov, B.N. Khoromskij, I.V. Oseledets, E.E. Tyrtyshnikov, A reciprocal preconditioner for structured matrices arising from elliptic problems with jumping coefficients, Linear Algebra Appl. 436 (2012) 2980–3007.

[10] S. Dolgov, R. Scheichl, A hybrid Alternating Least Squares – TT Cross algorithm for parametric PDEs, arXiv preprint 1707.04562, 2017.

[11] S.V. Dolgov, TT-GMRES: solution to a linear system in the structured tensor format, Russ. J. Numer. Anal. Math. Model. 28 (2013) 149–172.

[12] S.V. Dolgov, B.N. Khoromskij, D.V. Savostyanov, Superfast Fourier transform using QTT approximation, J. Fourier Anal. Appl. 18 (2012) 915–953.

[13] S.V. Dolgov, D.V. Savostyanov, Alternating minimal energy methods for linear systems in higher dimensions, SIAM J. Sci. Comput. 36 (2014) A2248–A2271.

[14] M. Eigel, C.J. Gittelson, C. Schwab, E. Zander, Adaptive stochastic galerkin FEM, Comp. Methods Appl. Mech. Eng. 270 (2014) 247 – 269.

[15] M. Espig, W. Hackbusch, A. Litvinenko, H. Matthies, E. Zander, Efficient analysis of high dimensional data in tensor formats, in: Sparse Grids and Applications, Springer, 2013, pp. 31–56.

[16] G.S. Fishman, Monte Carlo: Concepts, Algorithms, and Applications, volume 1196, Springer New York, 1996.

[17] I. Graham, F. Kuo, D. Nuyens, R. Scheichl, I. Sloan, Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications, J. Comput. Phys. 230 (2011) 3668–3694.

[18] L. Grasedyck, D. Kressner, C. Tobler, A literature survey of low-rank tensor approximation techniques, GAMM-Mitt. 36 (2013) 53–78.

[19] W. Hackbusch, Tensor Spaces And Numerical Tensor Calculus, Springer–Verlag, Berlin, 2012.

[20] A.S. Householder, The Theory Of Matrices In Numerical Analysis, Introductions To Higher Mathematics, 1964.

[21] V. Kazeev, B. Khoromskij, E. Tyrtyshnikov, Multilevel Toeplitz matrices generated by tensor-structured vectors and convolution with logarithmic complexity, SIAM J. Sci. Comput. 35 (2013) A1511–A1536.

[22] V.A. Kazeev, B.N. Khoromskij, Low-rank explicit QTT representation of the Laplace operator and its inverse, SIAM J. Matrix Anal. Appl. 33 (2012) 742–758.

[23] V. Khoromskaia, D. Andrae, B. Khoromskij, Fast and accurate 3D tensor calculation of the fock operator in a general basis, Comput. Phys. Commun. 183 (2012) 2392 – 2404.

[24] V. Khoromskaia, B. Khoromskij, R. Schneider, Tensor-structured factorized calculation of two-electron integrals in a general basis, SIAM Journal on Scientific Computing 35 (2013) A987–A1010.

[25] V. Khoromskaia, B.N. Khoromskij, Grid-based lattice summation of electrostatic potentials by assembled rank-structured tensor approximation, Comput. Phys. Commun. 185 (2014) 3162 – 3174.

[26] B. Khoromskij, C. Schwab, Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs, SIAM J. Sci. Comput. 33 (2011) 1–25.

[27] B.N. Khoromskij, Tensor-structured preconditioners and approximate inverse of elliptic operators in $\mathbb{R}^{\mathtt{d}}$, Constr. Approx. 30 (2009) 599–620.

[28] B.N. Khoromskij, $\mathcal{O}(\mathtt{d}\log\mathtt{n})$–Quantics approximation of $\mathtt{N}$–$\mathtt{d}$ tensors in high-dimensional numerical modeling, Constr. Approx. 34 (2011) 257–280.

[29] B.N. Khoromskij, Tensor numerical methods for multidimensional PDEs: theoretical analysis and initial applications, ESAIM: Proc. 48 (2015) 1–28.

[30] B.N. Khoromskij, V. Khoromskaia, H.J. Flad., Numerical solution of the Hartree–Fock equation in multilevel tensor-structured format, SIAM J. Sci. Comput. 33 (2011) 45–65.

[31] B.N. Khoromskij, I.V. Oseledets, DMRG+QTT approach to computation of the ground state for the molecular Schrödinger operator, Preprint 69, MPI MIS, Leipzig, 2010.

[32] B.N. Khoromskij, I.V. Oseledets, Quantics-TT collocation approximation of parameter-dependent and stochastic elliptic PDEs, Comput. Methods Appl. Math. 10 (2010) 376–394.

[33] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (2009) 455–500.

[34] D. Kressner, C. Tobler, Low-rank tensor Krylov subspace methods for parametrized linear systems, SIAM J. Matrix Anal. Appl. 32 (2011) 273–290.

[35] H. Matthies, A. Keese, Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations, Comput. Methods Appl. Mech. Eng. 194 (2005) 1295–1331.

[36] H. Matthies, E. Zander, Solving stochastic systems with low-rank tensor compression, Linear Algebra Appl. 436 (2012) 3819–3838.

[37] F. Nobile, R. Tempone, C. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, SIAM J. Numer. Anal. 46 (2008) 2309–2345.

[38] I.V. Oseledets, Tensors inside of matrices give logarithmic complexity, Preprint 2009-04, INM RAS, Moscow, 2009.

[39] I.V. Oseledets, Approximation of $2^d \times 2^d$ matrices using tensor decomposition, SIAM J. Matrix Anal. Appl. 31 (2010) 2130–2145.

[40] I.V. Oseledets, DMRG approach to fast linear algebra in the TT–format, Comput. Meth. Appl. Math. 11 (2011) 382–393.

[41] I.V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. 33 (2011) 2295–2317.

[42] I.V. Oseledets, Constructive representation of functions in low-rank tensor formats, Constr. Approx. 37 (2013) 1–18.

[43] I.V. Oseledets, S.V. Dolgov, Solution of linear systems and matrix inversion in the TT-format, SIAM J. Sci. Comput. 34 (2012) A2718–A2739.

[44] D.V. Savostyanov, QTT-rank-one vectors with QTT-rank-one and full-rank Fourier images, Linear Algebra Appl. 436 (2012) 3215–3224.

[45] J. Sherman, W.J. Morrison, Adjustment of an inverse matrix corresponding to a change in one element of a given matrix, Ann. Math. Stat. 21 (1950) 124–127.

[46] V. Simoncini, D.B. Szyld, Theory of inexact Krylov subspace methods and applications to scientific computing, SIAM J. Sci. Comput. 25 (2003) 454–477.

[47] A. Teckentrup, R. Scheichl, M. Giles, E. Ullmann, Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients, Numer. Math. 125 (2013) 569–600.

[48] R.A. Todor, C. Schwab, Convergence rates for sparse chaos approximations of elliptic problems with stochastic coefficients, IMA J. Numer. Anal. 27 (2007) 232–261.